DIME DYNAMIC DOCUMENTATION TRAINING
Exercise 2

Luiza Andrade & Mrijan Rimal

June 9, 2017

Contents

Introduction

Exercise 1 introduced you to the basics of how to import tables and figures to a LaTeX document. This exercise will introduce you to some intermediate topics commonly used to make your document look even more professional.

We will also show how LaTeX can be used to create a dynamic document that updates automatically once your output from, for example, Stata or R is updated without any error prone manual copy-and-pasting.

1 Exporting tables

In this section, you have the option of using the template do file provided in Dynamic-Documentation/Exercises/Stata Export Exercise/Export tables and images.do to export graphs and figures OR using your own data and generating tables and graphs that can be imported into LaTeX.

If you're using the do file provided, change the folder paths in the do file to your directory structure and run the do-file. This will export two tables and two graphs to the Raw folder that will be used in the exercises that follow.

If you would like to work with your own data, please follow the following steps.

- Make sure that you export **two tables** and **two graphs** using your own data. For help, you can look at the folder called "Exercise Stata - How to export tables and graph from Stata to LaTeX".
- Make sure that all the exporting is done using Stata do-files, or R scripts so you can easily export them again by just running the code. Towards the end of this exercise we will ask you to make changes to your do-file or R script so if you are using a file from an actual project you may want to make a copy of your do-file or R script that you can use in this exercise.
- Import all the tables and figures you have created in step 1 into your LaTeX document. Sample code to import tables and figures can be found in the **DIME Templates** sub-folder. Make sure to use the code in `Template 1 - Importing tables.tex` file to import tables (not graphs), and the code in `Template 2 - Importing figures.tex` file template to import graphs (not tables). It will not work if it is the other way around.
- Make sure you give a caption to your tables.

2 Intermediate LaTeX Exercises

2.1 Adding Sections to your document

LaTeX automatically formats the document and different section headers and subheaders according to pre-defined formats. It also allows you to automatically create a `Table of Contents` based on what has been defined as sections and subsections.

Sections can be created using `\section{title}` command. LaTeX automatically numbers all the sections in the order you put them. Since you don't manually specify chapter and section numbers, you can cut and paste subsections from the end to the beginning and their numbering will update automatically!

Similarly, subsections can be created using `\subsection{title}` and sub-sub-sections can be created using `\subsubsection{title}`. The subsections will be numbered on the format 1.1 and the subsubsection will be numbered on the format 1.1.1.

Task 1: Create two different sections in your LaTeX document, one called *Tables* and other called *Figures*.

Note: Sections and subsections can be created using `\section*{title}` *if you do not want your sections numbered in the document. However, these sections and subsections will not be shown in the table of contents.*

2.2 Adding Table of Contents

After we have set up sections, sub-sections and sub-sub-sections you can easily add a table of contents to your LaTeX document by typing `\tableofcontents` in your document. You can create this anywhere you want, but it is typically created right after `\begin{document}` – or `\maketitle` if you have a title. An example is shown below:

```
\documentclass[12pts]{article}


\title{My Awesome Document}

\author{John Doe}


\begin{document}

\maketitle

\tableofcontents

\newpage
```

The above LaTeX code is used to generate table of contents below.

My Awesome Document

John Doe

May 31, 2017

Contents

Figure 1: Example table of contents generated from our document in Annex 1.

Task 2: Add a table of contents to the first page of your LaTeX document.

2.3 Referring to your tables, and figures in a document

Just like the numbering of section and sub-sections, LaTeX uses a dynamic referencing system for tables and figures etc. In part of your text where you describe tables and figures you are likely to want to refer to them on a format similar to *"As you can see in figure 2..."*. Since the numbering of tables and figures is updated automatically, you need a way for your references in your text to be updated automatically as well.

In LaTeX that is solved by giving tables and graphs a unique name using the `\label{}` command. In the text where you want to reference a figure or a table, you reference the name used in the label and LaTeX will update the numbering for you as your documents grows and changes.

The labels are categorized into the type of item you are referencing, so you label tables and figures slightly differntly. We will show how it is done in the following sub-sections.

Task 3: Below the first figure of your file, write a short paragraph describing it.

2.3.1 Referring to Figures in a document

To refer to figures, we will use the `\label{fig:figurename}` command. Inside the brackets in the `\label{}` command you see two parts seperated by a colon. The first part (`fig`) indicates that this is a figure that we are assigning a label to. The second part (`figurename`) should be replaced with the unique name you want to use to refer to this figure.

We must add the `\label{fig:figurename}` line inside the figure environment in our codes, and below the caption. Otherwise, it will be not be displayed correctly. Here's an example:

```
\begin{figure}[H]
        \includegraphics[width=\textwidth]{../Raw/iegraph.png}
        \caption{Add figure title here}
        \label{fig:iegraph}
\end{figure}
```

Now you can type `\ref{fig:iegraph}` to refer to this figure and it will be automaticallyupdated using the correct number in the document. In our example, if the `iegraph` figure is the fourth one in the document, `As you can see in figure \ref{fig:iegraph}` will be displayed as *As you can see in figure 4* in the document.

Task 4: Edit the paragraph you wrote below figure 1 to include a dynamic referrence to it. Use *Build & View* to check that it works.

Task 5: Change the order of the two figures in your document and make what used to be the second figure the first one. Use *Build & View* to update your PDF and make sure that the figure reference was updated.

2.3.2 Referring to Tables in a document

Similarly for tables, we can also refer to them anywhere in the document using the same method. To refer to tables, we will use `\label{tab:tablename}` command inside the table environment and below the caption, as in the example below.

```
\begin{table}[H]
        \caption{Add a title to this table}
        \input{../Raw/sample_sizes.tex}
        \label{tab:samplesize}
\end{table}
```

2.4 Text Formatting

Now that you have created sections in your document, and added text, you might want to change some of the formatting of your text. In LaTeX you format your text using packages and code. We will here include a few topics.

2.4.1 Bold, italic and underlined

You can make part of your text bold using `\textbf{text}`. To underline part of your text, use `\underline{text}`. To make part of your text italic use `\textit{text}`. Here's how it looks:

`\textbf{This is a bold text.}`	**This is a bold text.**
`\underline{This is an underlined text.}`	<u>This is an underlined text.</u>
`\textit{This is an italic text.}`	*This is an italic text.*

Task 6: Choose some words in your text paragraph to be turned into bold, italic and underline. If you're using TeXStudio, you can use the shortcuts `CTRL+B` for bold and `CTRL+I` for italic.

2.4.2 Text color

Font color can be changed using `\textcolor{color}{text}`. You can find a list of color options in the LaTeX WikiBook, but here are a few examples:

`\textcolor{red}{This is a read text.}`	This is a read text.
`\textcolor{Gray}{This is a gray text.}`	This is a gray text.
`\textcolor{Cyan}{This is a Cyan text.}`	This is a Cyan text.
`\textcolor{RoyalPurple}{This is a RoyalPurple text.}`	This is a RoyalPurple text.

Task 7: Change the color of a sentence in your paragraph. Try combining `\textcolor` with `textbf`.

2.5 Making a Dynamic Document

Here, we will produce a dynamic document. Please only do this do **if** you have completed all tasks up to `Part 2.5` of this exercise document and generated a document. This part will show you how easily LaTeX is updating your formatted documents without having to make any manual changes or manually copy and paste any files, code or results.

1. Make sure that the you have generated a PDF document by compiling the your LaTeX code. Now copy the PDF created to a separate location from where LaTeX is exporting it. You will only need this second copy of your PDF document temporarily, so you can copy it to your desktop for example.

2. Make a significant change to your dataset by, for example, dropping a large subset of your data set. You want to make such a large change to your dataset that when you re-run your do-file or your R script, your outputs should look very different.

   If you are using the Stata do-file template provided in the Dynamic Documentation Folder, then only keep observations with population growth more than 0. This can be done by using the Stata codes `drop if popgrowth < 0`.

3. Rerun the do-file or R script you used to generate the tables and files you imported to LaTeX in this exercise. Make sure to not change any file paths to where your tables and graphs were exported. You want your own tables and graphs to be replaced and overwritten.

4. Now, go back to the LaTeX code where you have imported the tables and graphs for this exercise and generate the pdf document again. Press *Build and Compile* under the *Tools*.

5. Now if you compare the PDF file you have just generated with you new data with the one you saved in a different folder in the beginning of this section, you will find that the tables have updated automatically

without you having to do any manual edits.

In this part you updated your tables and figures in the final document without having to copy any results or files manually. Such manual updates are by far the most common reason for errors while generating and sharing research results. This is the most important reason why we at DIME are promoting LATEX, as manual errors cause us to lose a lot of time and in the worst case leads to mistakes in results that are published.

3   Additional LATEX features

3.1   Line Spacing

It is common that different publication standards require different line spacing. This can be achieved using the `setspace` package.

To use this package you need to start by importing it. To import the package you add `\usepackage{setspace}` next to where a lot of other packages are imported using the `\usepackage{}` command. Next you simply tell LATEX which line spacing you want and you need to do this before `begin{document}`. `\singlespacing` makes LATEX use single line space throughout the document. Similarly, `\doublespacing` and `\onehalfspacing` set the document to double line space and one half line space respectively. See example below:

```
\usepackage{setspace}
\doublespacing


\begin{document}
```

3.2   Rotating a table

Sometimes the tables are very wide and need to be in landscape format. This can be adjusted using the `adjustbox` package which we have been using for importing tables. To rotate the table to landscape, we can use the `adjustbox` feature with the `angle = 90` option as shown in blue in the code below.

```
\begin{table}[H]

        \caption{Add table title}

        \begin{adjustbox}{angle = 90}

                \input{sample_sizes.tex}

        \end{adjustbox}

        \label{tab:samplesize}

\end{table}
```
This will rotate the table to the specified degree in the final document.

3.3   Creating items

As you have probably noticed in this document, it is also possible to make lists. The `enumerate` environment created numbered lists, while the `itemize` environment only creates items.

To make a list, you must first create the environment by typing `\begin{itemize}` and `\end{itemize}`, then type `\item` before each of the items in your list, as shown below:

```
Here's a numbered list:

\begin{enumerate}

\item Item one

\item Item two

\item Item three

\end{enumerate}
```

Here's a numbered list:

1. Item one

2. Item two

3. Item three

```
Here's an unnumbered list:

\begin{itemize}

\item Item one

\item Item two

\item Item three

\end{itemize}
```

Here's an unnumbered list:

- Item one

- Item two

- Item three